DOCUMENT RESUME

ED 139 105 EA 009 527

AUTHOR Wulfeck, Wallace H., II
TITLE An Algorithmic Approach to Curriculum Construction: Development, Computer Implementation, and Evaluation of a Method for Identifying Instructional Content Sequences.
PUB DATE Apr 77
NOTE 11p.; Paper presented at the Annual Meeting of the American Educational Research Association (New York, N.Y., April 4-8, 1977)

EDRS PRICE MF-$0.83 HC-$1.67 Plus Postage.
DESCRIPTORS *Algorithms; Computer Science; Curriculum Design; Educational Research; *Geometry; *Learning Processes; Memory; *Problem Solving; *Simulation

ABSTRACT

This paper describes an investigation of problem sequencing as part of a symposium presenting theoretical concerns and research findings regarding several diverse types of strategies for designing and presenting instructional materials. The first step involved analyzing a set of problems, to determine the knowledge (rules) required for solving the problems. A set of geometric construction problems was analyzed using a method for deriving and simplifying rules or algorithms for solving the problems. Next, assumptions were adopted concerning how subjects learn and apply rules to solve problems. A computer was programmed according to these assumptions, and given some of the simplified rules identified during the previous step, in order to simulate some aspects of subjects' problem-solving and learning. Various problem sequences were then given to the program, to identify those that were solvable and "learnable." Finally, for initial evaluation of the approach, sequences that were learnable for the program were given to subjects, whose performance was compared with the performance of subjects given random or learner-controlled sequences. (Author/MLF)

Session 16.03

An Algorithmic Approach to Curriculum Construction: Development, Computer Implementation, and Evaluation of a Method for Identifying Instructional Content Sequences.*1

Wallace H. Wulfeck II
NAVY PERSONNEL RESEARCH and DEVELOPMENT CENTER
San Diego, CA 92152

Introduction.

This paper describes an investigation, based on the Structural Learning Theory (Scandura, 1973; in press), of problem sequencing. The first step involved analyzing a set of problems, to determine the knowledge (rules) required for solving the problems. Here, a set of sample problems (geometric construction problems in this study) was analyzed using a method (Scandura, Durnin, and Wulfeck, 1974) for deriving and simplifying rules or algorithms for solving the problems. Next, assumptions were adopted from the Structural Learning Theory concerning how subjects ~~learn and apply rules to solve problems~~. A computer was programmed according to these assumptions, and given some of the simplified rules identified during the previous step, in order to simulate some aspects of subjects' problem solving and learning. Various problem sequences were then given to the program, to identify those that were solvable and "learnable." Finally, for initial evaluation of the approach, sequences which were learnable for the program were given to subjects, whose performance was compared with the performance of subjects given random or learner-controlled sequences.

Analysis of Geometric Construction Problems.

Scandura, Durnin, and Wulfeck (1974) developed a quasi-systematic method for constructing sets of rules underlying successful solutions for problems.*2 Briefly, it involves: (1) sampling and classifying problems to be analyzed into groups of

---

*1. This paper is based on sections of the author's dissertation (Wulfeck, 1975) conducted under the chairmanship of Dr. J. M. Scandura. This research was supported in part by a Dissertation Year Scholarship from the University of Pennsylvania, in part by a grant from the Office of Computing Services, University of Pennsylvania, and in part by a grant from the National Institute of Health to Dr. J. M. Scandura.

The author is now at the Navy Personnel Research and Development Center, San Diego, CA 92152. The opinions and assertions contained herein are those of the writer and are not to be construed as official or reflecting the views of the Navy Department.

*2. Theoretical foundations underlying the method of analysis are given in Scandura (1973, Chap. 5).

similar problems; (2) specifying a rule for solving each classified problem; (3) identifying lower order (component) rules which may be operated upon (e.g. concatenated) to form the solution rules identified in step 2, and, more importantly, identifying higher order rules which operate on the lower order rules to generate the solution rules; (4) eliminating the solution rules made unnecessary by the introduction of the lower and higher order rules; and (5) testing and refining the resulting set of rules so that solution rules for all the originally grouped problems, and new problems from the same groups, can be generated. Requiring that the resulting set of rules account for problem solutions in this way makes the analysis "self-validating."

To illustrate the method of analysis, consider the following two geometric construction tasks:

(1.) "Given point P, Line L, and Distance D, construct a circle with radius D, passing through P, tangent to L."

(2.) "Given triangle ABC, construct (circumscribe) a circle passing through points A, B, and C."

After specifying solution rules for these tasks, and others, lower order rules involved in generating the solutions were identified:

R1: Construct a circle with a given point as center, and a given distance as radius. (p, d) ==R1=> c(p, d)

R2: Construct a line parallel to a given line at a given distance from the given line. (l, d) ==R2=> l∥(l, d)

R3: Construct the locus of points (a line) equidistant from two given points. (p, p′) ==R3=> l⊥(p, p′)

R4: Construct a circle with a given point as center, passing through another given point. (p, p′) ==R4=> c(p, d(p, p′))

From commonalities among solution rules for tasks (1) and (2) above, and others, Scandura, Durnin, and Wulfeck (1974) identified a "two-loci" higher order rule which could operate on rules like R1 - R4 above to yield solution rules for tasks (1) and (2). The higher order rule essentially concatenates rules for constructing two different loci, then a rule for constructing the "goal figure" from an intersection point (px) of the loci. For tasks (1) and (2), the output rules are [R1(p, d), R2(l, d), R1(px, d)], and [R3(a, b), R3(b, c), R4(px, a)].

Rule R1 above is "basic" enough so that it need not be further analyzed. However, the others (including the higher order rule) can be further analyzed using the same method. For example, rules R2 and R3, when re-analyzed, were found to be

3

generable from the following lower order rules:

R5: Construct a point at a given distance from a given line.

R6: Construct a point equidistant from two given points.

R7: Construct a line through two given points.

A higher order rule which concatenates two rules for constructing points satisfying the same condition a required line must satisfy, then a rule for constructing the "goal figure" (the line) can operate on rules R5 - R7 to yield rules R2 and R3 as outputs.

The two higher order rules discussed above can also be further analyzed. Both of them involve concatenating two rules which yield separate elements, then reconcatenating a rule which operates on the separate elements to yield a "goal figure" (see Wulfeck, 1975, for details of the analysis with respect to higher order rules). In effect, higher order rules can themselves be concatenated using a higher order concatenation rules.

In extending the geometric construction anayses, the method of analysis described above was reapplied "recursively" as illustrated. Analysis continued until rules performable by seventh-grade students were reached. The basic rules identified included lower-order rules for constructing a circle given the center and radius, drawing a line or segment through two given points, finding a point of intersection of intersecting lines or circles, measuring (setting a compass to) the distance between given points, and choosing arbitrary points or distances. The basic higher order rules included "composition" (concatenate two rules when the output of one is a required input of the second), and "conjunction" (combine two rules yielding individual elements into a single rule for constructing two elements).

At this point, it might appear that to derive sequences, one need only trace backward through the analysis which yielded the basic set. However, this strategy is insufficient since (a) there are other sequences than the reverse of the order in which problems were considered, and (b) there are sequences which include problems not even considered during the original analysis, but solvable via the identified rules. In general, the competence analysis does not yield a strict hierarchy from which it is possible to "read" performable sequences. Furthermore, and most important, one cannot make predictions about problem solving and learning with respect to task sequences without making (explicit or implicit) assumptions about human learning and performance.

Computer Implementation.

This approach to sequence investigation therefore depended upon adoption of explicit assumptions concerning subjects' application and acquisition of rules in problem solving situations. The simplest of several possible rule control mechanisms (sets of problem solving and learning assumptions) proposed by Scandura (1973) was adopted here, and implemented as a computer program to aid in sequence development.

It is assumed that the mechanism is goal directed, and that problems are first uniformly and correctly interpreted (as a pair consisting of given information [S], and a unitary goal [G], not a series of subgoals).*3

INSERT FIGURE 1 ABOUT HERE.

Briefly, the program works as shown in Figure 1: (1) Given a problem situation <S, G> for which at least one "applicable" rule is available (in the set of rules available to the program), one of these is applied to S to satisfy G. In the program, applicability decisions are made by "matching" specifications of rule ranges and domains against G and S. (2) Given a goal situation for which no appliable rules are available, then control shifts to the higher level goal H(G) of deriving a rule applicable to the previous goal. (3) Given that H(G) is satisfied, then any derived rule is added to the set of available rules, and control reverts to the previous goal ~~H(~~ G ~~)~~.

These control assumptions are intentionally neutral with respect to other limitations on human information processing capacity, memory, etc. That is, the program is built to work perfectly with respect to applicability determination, rule execution, access of the rule set, etc. Under these assumptions, the program is an idealized problem solver; human subjects, of course, are not. To limit the program's performance for simulation purposes (in the absence of more formal limiting assumptions), a goal limit (GLIMIT in Figure 1) is imposed which controls the highest level goal, per problem, the mechanism is allowed to attempt.*4

---

*3. In the Structural Learning Theory, it is assumed that a presented problem is adopted as a pair of ordered subgoals, "interpret the problem" (determine its meaning), and "solve the interpreted problem." While this paper is concerned with the second of these, no essential changes in methods or approach appear necessary to treat the first (see Scandura, 1973; in press). Some directions in investigating interpretation of geometric construction problems are given in Wulfeck (1975).

The overall across-problem operation of the program is detailed in Figure 2. (This part of the program is concerned with problem "book-keeping" and is outside the theory.)

INSERT FIGURE 2 ABOUT HERE.

The program is initially given a set of basic rules; this is the only part of the program specific to the particular content domain. When the program is given a list of problems to solve, it attempts problems in turn, discarding solved problems but adding rules derived during those solutions to its rule set. Failed problems are retained on a list, and are later re-attempted. This continues until all problems are solved (or until the number of failed problems reaches some prespecified failure limit). This process has the effect of re-ordering presented problems into a sequence in which each problem is solvable according to control assumptions and the goal limit constraint on its first presentation. The program outputs may then be used to discard redundant problems, to rearrange problems, or to add intermediate problems so that some unsolved problem(s) may become solvable.

The highest level goal at which a problem becomes solvable by the program (before goal reversion) seems to provide a natural measure of the "step size" or "difficulty ~~level~~" of (between) sequenced problems. For example, problems for which a solution rule already exists in the current rule set, are solvable at the initial goal level (CG = 1 in Figure 1). Problems whose solution rules are derivable from rules in the set via a higher order rule in the set are solvable at the second goal level (CG = 2), etc. This step size measure summarizes (but is not perfectly or linearly related to) both the amount of processing problem solutions entail, and the number of rules involved in generating problem solutions. The central point is that there is no fixed a priori step size for any problem; the step size is always relative to the knowledge base--the rule set--existing at the time a problem is presented. When the goal limit (GLIMIT in Figure 1) is set to some number N, this restricts the output sequences of the program to those in which every problem is solvable at CG $\leq$ N; sequenced problems are thus restricted to some maximum step size or level of difficulty.

*4. Extensions of the mechanism, designed to deal with memory processes, processing capacity, and other capabilities of subjects are available in the Structural Learning Theory. See Scandura (1973, in press).

Empirical Evaluation.

According to the above discussion, we expect that with human subjects, both failure frequencies and solution latencies are related to increasing step size. Then, sequences in which step sizes are kept small should lead to better overall performance than those in which step size is uncontrolled.

Four groups (ten seventh-grade subjects each) were given different sequence arrangements of geometric construction tasks: Group X1 received a sequence of 20 problems devised, using the program, so that the step size for problems 18 and 20 was three, for all others, two. Group X2 received a sequence obtained from the first by deleting four problems. According to the program's performance, this increased the step sizes for problems 9 and 13 to three, and decreased the step size for problem 18 to two. (The higher order rule for problem 9 derived at CG = 3 was usable on problem 18). Group R was given the original 20 problems, but after problem 6, problem order was random. Subjects in group L, after problem 6, were allowed to choose each problem to attempt next.*5

Results and Discussion.

Means of subjects' percent success on problems attempted after problem 6 were: group X1, 85%; X2, 73%; L, 47%; and R, 38%. All differences were significant ($p < .05$) except X1-X2 and L-R. Evidently, sequences derived according to the program so that step sizes are uniformly small lead to significantly better performance than do random or learner-controlled sequences. Also, subjects must have used previously derived rules in generating solutions to later problems, otherwise all groups would have performed similarly.

Mean times to solution on problems common to groups X1 and

---

*5. Procedures: Subjects were individually given problems one at a time (except group L after problem 6) on separate pages. For each problem, the problem statement was read aloud by the experimenter, given elements were pointed out, relational terms were explained, and a sketch of the "goal figure" in required relations to given elements was drawn. Subjects performed all constructions on the problem pages. If a problem was failed, the subject was shown a solution rule for it, and was required to execute the rule correctly on the problem page. Subjects retained their problem pages through their sequences for memory support of previously derived solution rules. Problems 1 through 6 were used to pretest subjects' prior knowledge: All subjects were initially given problem 6; no subject solved it. Subjects were then given problems 1 through 6, and all subjects solved problems 1 through 6. These results supported the assumption that basic rules were at an appropriate level of detail.

X2 were in complete agreement with predictions from the Structural Learning Theory. The only significant differences (in mean log solution times) across X1 and X2 occurred in predicted directions on problems 9 (X1<X2), 13 (X1<X2), and 18 (X2<X1). These differences clearly track the step size differences of problems, and therefore provide strong additional support for the viability of a goal-switching type of control mechanism. Also, X2 subjects were evidently able to retain and use higher order rules derived on some problems for later problems even though they were not given memory support for higher order rules.

As expected, a significant positive association between step size and frequency of failure occurred over the experiment. No subject solved a problem when the step size was greater than three. Perhaps "memory load" approaches subjects' processing capacities at step sizes around three, and if so, this might provide a more appropriate constraint on the program's performance than simply imposing a goal limit.*6

However, a significant step size (2 or 3) by sequence condition interaction on percent success scores also occurred, such that L, and particularly R subjects performed differentially more poorly on step-size-three problems than did the X1 or X2 subjects. While other factors (e.g. motivation) may be involved, the X1 and X2 sequences have a "chaining" property such that problems' solution rules are often derived using rules from recent previous problems. This was less often the case in the L and R sequences. Chaining is only indirectly related to controlled step size: when a relatively small set of problems is given to the program to be sequenced, the restrictions on goal level force a moderate degree of chaining. On the other hand, if a larger set of problems, including many different alternative intermediate problems between "basic" and terminal ones, were given to the program, controlled step size would not necessarily force much chaining. To the extent the rule "recency" or <u>einstellung</u> are involved in subjects' memorial processes, it may be desirable to include additional assumptions concerning memory mechanisms.

There was a fairly wide range over group L subjects of success on chosen problems. Some L subjects had explicit bases for choosing "next" problems, which seemed related to the step sizes of chosen problems, and to success on problems: Three subjects who stated that they chose on the basis of similarity (of problem statement and display) to previous problems, chose chained problems of small step size (most often two), and solved about 73% of these. However, two subjects, who chose "dissimilar" problems,

---

*6. Determining memory load involves considering goal levels, and numbers of rules involved at different levels, but also the loads imposed by particular rules during their execution. See Scandura (1973; in press).

never chose problems with step sizes as small as two, and solved none of their chosen problems. The remaining subjects indicated no particular basis for selection, and solved about half their chosen problems. These results, and others (e.g. Pask and Scott, 1971) suggest that some subjects may have useful problem selection skills (rules), toward which additional research might be directed.

REFERENCES

Pask, G. and Scott, B. C. E. Learning and teaching strategies in a transformational skill. British Journal of Mathematical and Statistical Psychology, 1971, 24, 205-229.

Scandura, J. M. Structural learning I: Theory and research. London/New York: Gordon and Breach, 1973.

Scandura, J. M. Problem solving: A structural/process approach with educational implications. New York: Academic Press, in press.

Scandura, J. M., Durnin, J. H. & Wulfeck, W. H. II Higher order rule characterization of heuristics for compass and straightedge constructions in geometry. Artificial Intelligence, 1974, 5, 149-183.

Wulfeck, W. H. II An algorithmic approach to curriculum construction: Development, computer implementation, and evaluation of a method for identifying instructional content sequences. Dissertation, University of Pennsylvania, 1975. Dissertation Abstracts International, 1976, 36(12), pt. 1, 7967A.

Figure 1.: Problem Solving and Learning Mechanism:

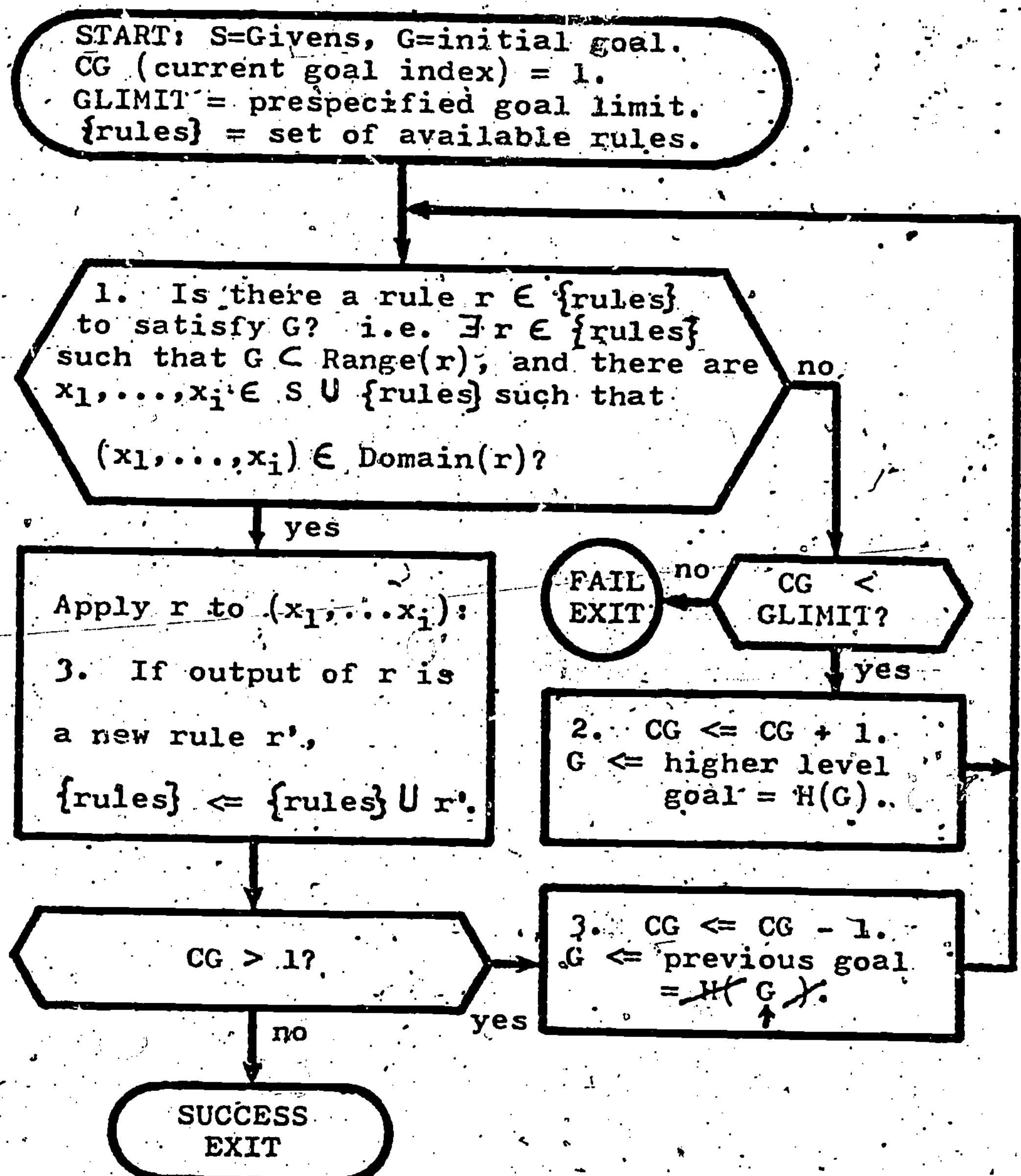

Figure 2.: Overall Program Operation